Preventive maintenance for heterogeneous industrial vehicles with incomplete usage data

Preventive maintenance for heterogeneous industrial vehicles with incomplete usage data

Dena Markudova, Sachit Mishra, Luca Cagliero, Luca Vassio, Marco Mellia, Elena Baralis

Politecnico di Torino,
Corso Duca degli Abruzzi 24, 10129, Turin, Italy

Lucia Salvatori, Riccardo Loti

Tierra SpA,
Corso Ferrucci 112, 10138, Turin, Italy

Abstract

Large fleets of industrial and construction vehicles require periodic maintenance activities. Scheduling these operations is potentially challenging because the optimal timeline depends on the vehicle characteristics and usage. This paper studies a real industrial case study, where a company providing telematics services supports fleet managers in scheduling maintenance operations of about 2 000 construction vehicles of various types. The heterogeneity of the fleet and the availability of historical data fosters the use of data-driven solutions based on Machine Learning techniques. The paper addresses the learning of per-vehicle predictors aimed at forecasting the next-day utilisation level and the remaining time until the next maintenance. We explore the performance of both linear and non-liner models, showing that machine learning models are able to capture the underlying trends describing non-stationary vehicle usage patterns. We also explicitly consider the lack of data for vehicles that have been recently added to the fleet. Results show that the availability of even a limited portion of past utilisation levels enables

Email addresses: `dena.markudova@polito.it` (Dena Markudova), `sachit.mishra@imdea.org` (Sachit Mishra), `luca.cagliero@polito.it` (Luca Cagliero), `luca.vassio@polito.it` (Luca Vassio), `marco.mellia@polito.it` (Marco Mellia), `elena.baralis@polito.it` (Elena Baralis), `lsalvatori@topcon.com` (Lucia Salvatori), `rloti@tierratelematics.com` (Riccardo Loti)

the identification of vehicles with similar usage trends and the opportunistic reuse of their historical data.

Keywords: Preventive Maintenance, Industrial Vehicles, Fleet management, Machine Learning, Classification

1. Introduction

Industrial vehicle fleets often need maintenance operations that are not trivial to plan. To ensure a properly functioning fleet, managers resort to periodic maintenance activities, which are subject to uncertainty and depend very much on the vehicle workload. This workload is also variable, for example, some vehicles could remain unused for a relatively long period of time and then be moved to a construction site where they work in full capacity for many days or weeks. Moreover, industrial fleets are highly diversified in models and types, each showing heterogeneous usage patterns. Thus, fleet managers, to properly manage the working activities in construction sites, need automated decision support systems to optimize the schedule of these periodic maintenance operations (Crainic & Laporte, 1998).

The advent of Controller Area Network (CAN) bus technology (Johansson et al., 2005) has enabled the acquisition, collection, and processing of vehicle usage data. The CAN bus allows for communication among the electronic control unit devices on board the vehicle, giving direct access to various signals describing the vehicle state. Once collected in a centralised repository, CAN bus data can be conveniently analysed by means of data mining and machine learning (ML) techniques to design predictive maintenance solutions (Zhang et al., 2019). Previous studies related to CAN bus data analysis focused on (i) predicting the future utilisation level of a vehicle by means of classification and regression techniques (e.g., Perrotta et al. (2017); Markudova et al. (2019)), (ii) aggregating vehicles with similar characteristics using clustering techniques (e.g., Alonso de Armiño et al. (2019); Halim et al. (2016)), and (iii) identifying malfunctioning of specific vehicle components based on anomaly detection methods (e.g., Zhang et al. (2019); Li et al. (2019)). This paper belongs to category (i) since it focuses on predicting for each vehicle the next-day utilisation level and the remaining time until the next maintenance by means of regression techniques.

Predicting the future usage level of industrial vehicles has attracted the interest of the research community for many reasons. Firstly, the transporta-

tion of raw materials and commercial goods heavily rely on vehicle fleets, which require periodic maintenance. Planning maintenance actions based on the actual vehicle usage, instead of a predefined schedule, significantly improves the efficiency of the overall industrial process (Dalzochio et al., 2020). Secondly, efficiently planning complex industrial processes and vehicle routes requires the aid of automated decision support systems (Zhou et al., 2016). One of the main limitations of previous machine learning-based solutions is the need for a sufficient amount of per-vehicle usage data, which, in turn, hinders the use of ML-based solutions on new vehicles, or on vehicles for which the amount of acquired data is limited.

This paper presents the outcomes on a industrial research work carried out with a company providing telematics services to industrial and construction vehicle producers. The goal of the research is to provide fleet managers with a machine learning-based approach to optimise maintenance operations of vehicles of various types and working in different scenarios. Note that we target only periodic maintenance. Detecting and managing vehicle failures is out of the scope of the present work. Our approach relies on regression models trained on both vehicle data and contextual features. For each vehicle, we target the prediction of next-day utilisation level and remaining time until the next maintenance. To overcome the lack of training data for vehicles recently added to the fleet, we explore the use of historical data acquired from similar vehicles. Specifically, we compare the new vehicles utilisation history to those of similar vehicles already in the fleet and look for the best approximate motif (Mueen et al., 2009). We then opportunistically reuse regression models trained on the historical data of the correlated vehicles to make predictions for the new vehicles. To evaluate the applicability of the proposed solution to heterogeneous fleets, we explore the performance of both linear and non-liner models to a large set of heterogeneous vehicles. We validate our approach on data from 60 construction vehicles.

Results show that machine learning models are able to capture the underlying trends describing non-stationary vehicle usage patterns, reaching a relative error as low as 12% while predicting the next-day usage level and an average error of 2.4 days for the time to next maintenance. Furthermore, even when a very limited portion of past utilisation data is available, our proposed approach achieved fairly good prediction performance, e.g., for the time to next maintenance 3% average mean residual error when vehicle data about at least one maintenance cycle is available, and 18% error when vehicle data about at least half of a maintenance cycle is given. In light of

these results, the data owner (collecting telematics data from real industrial vehicles) has decided to put the present application under deployment, thus enabling further tests, optimisations, and extensions.

The rest of the paper is organised as follows. Section 2 overviews the related literature. Section 3 formalises the problem addressed in the paper. Section 4 describes the analysed dataset, while Section 5 describes the adopted data-driven methodologies. Section 6 summarises the main experimental results and, finally, Section 7 draws conclusions and discusses the future research directions.

2. Literature review

Recently, there have been several studies that analyse CAN bus data by means of supervised machine learning techniques. The common goal is to predict the values of the main vehicle usage indicators.

Vehicle usage level prediction. Perrotta et al. (2017) addressed the problem of predicting trucks fuel consumption by applying regression models such as Support Vector Machines, Random Forest, and Artificial Neural Networks. Beyond the fuel consumed, authors considered various other CAN bus data features (e.g., gross vehicle weight, the vehicle speed, average acceleration, use of brakes and acceleration pedal, travelled distance) as well as geographical information. Similar works have been presented by Almer (2015); Wickramanayake & Bandara (2016); Nguyen & Wilson (2010); Delgado et al. (2012). In most of the aforesaid studies, the authors carried out a correlation analysis of vehicle usage data sets to identify the most discriminating features to predict future vehicle usage. For example, they studied the influence of road, weather, and driver information on vehicle usage forecasts. Unlike the present work, most of the proposed solutions are tailored to specific vehicle types such as public buses (Wickramanayake & Bandara, 2016), waste collectors (Nguyen & Wilson, 2010), heavy duty trucks (Delgado et al., 2012). Conversely, this work addresses the analysis of a heterogeneous fleet of vehicles belonging to different types and models and summarises the results achieved in a selection of relevant case studies.

Preliminary attempts to predict the next-day utilisation level and the remaining time to maintenance have been made in (Markudova et al., 2019) and (Mishra et al., 2020), respectively. However, both the aforesaid studies assume that, for every vehicle, a sufficient amount of training data is

available. Furthermore, the predictions do not rely on multivariate models. This paper extends the aforesaid studies, giving particular attention to the analysis of new and semi-new vehicles, the parameter sensitivity, and the evaluation of more complex predictors.

Maintenance scheduling and resource allocation. The use of Big Data techniques and ML algorithms to process information relevant to maintenance activities, scheduling, and resource allocation is mainstream (Morariu et al., 2020). Specifically, data-driven vehicle maintenance planning has already been addressed using various optimisation methods. For example, in Rashidnejad et al. (2018) the authors have applied genetic algorithms to plan the maintenance of geographically distributed assets by considering routing constraints and travel time to reach the assets. Robert et al. (2018) presented a dynamic optimisation method to plan maintenance of heavy vehicles by jointly scheduling maintenance operations and production activities, whereas Mohamed et al. (2017) proposed a data-driven simulation framework for planning snow removal activities, considering weather and truck-related data acquired by sensors. All the aforesaid strategies can be supported by accurate predictions of the vehicle usage level.

Route optimisation. A parallel branch of research has been devoted to optimise activities of fleet vehicles, which is a priority in several industrial processes (Barreto et al., 2017). For example, the work presented by Hellstrom et al. (2009) aimed at minimising trucks fuel consumption by optimising routes. Similar analyses tried to combine CAN Bus data with trip information to analyse the routes travelled by cars (Zeng et al., 2015) and trucks (Caapraz et al., 2016), respectively. The study presented in this paper can be instrumental in planning vehicle routes, because the ML-based prediction outcomes are promptly usable by fleet managers, that can include vehicle maintenance in their route optimisation.

3. Problem statement

Our goal is to forecast the next-day utilisation level and the time remaining until the next maintenance for an arbitrary vehicle v. Notice the difference between calendar time (in C^v and D^v), measured in days, and utilisation time (in U^v, T^v and L^v), measured in hours.

Notation.

- N^v [days]: duration of the period in which historical usage data are available for vehicle v.
- $U^v(t)$ [hours]: utilisation time series of vehicle v. t is an independent variable measured in [days]. In our experiments the value range for t is a subset of days between the oldest date in our data (i.e., 2015-01-12) and the latest one (2018-10-09) of size N^v.
- T^v [hours]: total target utilisation time of v between two consecutive maintenance operations[1]. This is the utilisation time of a *cycle* between two maintenance operations.
- $C^v(t)$ [hours]: time already passed from the last maintenance operation of v.
- $L^v(t)$ [hours]: series of the utilisation times left to the next maintenance operation of v. On an arbitrary time t, it is computed as follows:

$$L^v(t) = T^v - \sum_{i=t-C^v(t)}^{t-1} U^v(i) \tag{1}$$

- $D^v(t)$ [days]: time series of the time left to the next maintenance of v.

Task formulation. On the current day t, our goal is to predict for vehicle v: (A) the next-day utilisation level $U^v(t+1)$, and (B) the number of days $D^v(t)$ left to the next maintenance.

According to the amount of available historical usage data, we classify vehicles in 3 different categories:

- *Old vehicle*: vehicle for which at least one maintenance cycle has already passed since data acquisition has started.
- *Semi-new vehicle*: vehicle for which the first maintenance cycle has not been completed yet, but data about at least half of the usage in one cycle ($\frac{T^v}{2}$) is already available.

[1]This time is assumed to be fixed for all the vehicles of the same type.

- *New vehicle*: vehicle that has been used for less than $\frac{T^v}{2}$ since the beginning of the data acquisition phase.

The aforesaid vehicle classification will be used to tailor the solutions to tasks A and B to different scenarios. Specifically, for task B we adopt three different approaches according to the vehicle category, whereas for task A we differentiate the approach used for old vehicles from those adopted for semi-new and new ones.

Objective functions. To effectively support fleet managers in planning periodic vehicle maintenance, the prediction systems aim at minimising the forecasting errors.

To assess the forecasting error while predicting the next-day utilisation level $U^v(t+1)$ on a sub-period of duration $N \leq N^v$, we adopt the absolute percentage error ($E^v_{\%}$) (Zaki & Meira, 2020). It accounts for the difference between the predicted ($U^v_{predict}$) and actual (U^v) utilisation levels on an arbitrary day t and is expressed as follows:

$$E^v_{\%} = 100 \cdot \frac{\sum_{t=1}^{N} |U^v_{predict}(t+1) - U^v(t+1)|}{\sum_{t=1}^{N} U^v(t+1)}$$

$E^v_{\%}$ is computed separately for each vehicle and indicates the mean absolute prediction error relative to the average usage of the vehicle itself. We then compute the Average Percentage Error, which indicates the average of $E^v_{\%}$ over all the vehicles used in the case study.

To assess the ability of the system to predict the remaining time to maintenance over a sub-period of duration $N \leq N^v$, we consider the following three objective functions: (i) *daily error* $E^v(t)$, (ii) *global error* E^v_{Global}, and (iii) *Mean Residual Error* $E^v_{MRE}(\widetilde{D})$. The daily error indicates the absolute gap between the predicted ($D^v_{Predict}$) and actual time (D^v) of the next maintenance on a day t:

$$E^v(t) = |D^v(t) - D^v_{Predict}(t)| \tag{2}$$

The global error is a mean of the daily errors over all the N predictions.

$$E^v_{Global} = \frac{\sum_{t=1}^{N} E^v(t)}{N} \tag{3}$$

The global error does not take into account the actual time left to maintenance. For example, an error of 1 day when we are close to the maintenance (e.g., $D^v(t) = 2$) is considered as equal as an error of 1 day when we are far from the maintenance (e.g., $D^v(t) = 100$). The mean residual error overcomes the above-mentioned issue by taking into account the actual closeness of the next maintenance cycle. Specifically, it averages the daily errors over a selection of critical days $\widetilde{D}$, which are the ones closer to the maintenance operation[2]. E^v_{MRE} is computed as follows:

$$E^v_{MRE}(\widetilde{D}) = \frac{\sum_{i:D^v(i)\in\widetilde{D}} E^v(i)}{i : D^v(i) \in \widetilde{D}} \tag{4}$$

The idea behind E^v_{MRE} is that fleet managers are mainly interested in getting accurate predictions when vehicles are towards the end of their maintenance cycle, i.e., when maintenance operations need to be scheduled soon.

The main objectives of our study can be summarised as follows: (i) predict the next-day utilisation level (task A) by minimising $E^v_{\%}$ and (ii) predict the remaining time to maintenance (task B) by minimising $E^v_{MRE}(\widetilde{D})$.

4. Data overview

4.1. Data description

Here we first analyse the historical usage data of industrial vehicles of various types. The data were provided thanks to a company offering telematics services to multiple vendors and were acquired through the CAN bus devices installed on-board the vendors' vehicles. Overall, we analysed data related to 2 239 vehicles belonging to 10 different types and located in 151 different countries spread all over the world. The dataset was acquired in a 4-year time period ranging from January 2015 to September 2018.

Vehicles are identified by a unique identifier (the *vehicle id*) and classified based on the *type* of construction vehicle (e.g., refuse compactor, single drum roller, paver). Each type then contains several *models* (i.e., a type subcategory), for which we can have different *units* in our dataset. For example, we have 44 different models of refuse compactors, 65 models of drum rollers, and 10 models of pavers.

[2]In the experiments, we focused on the last 29 days per maintenance cycle.

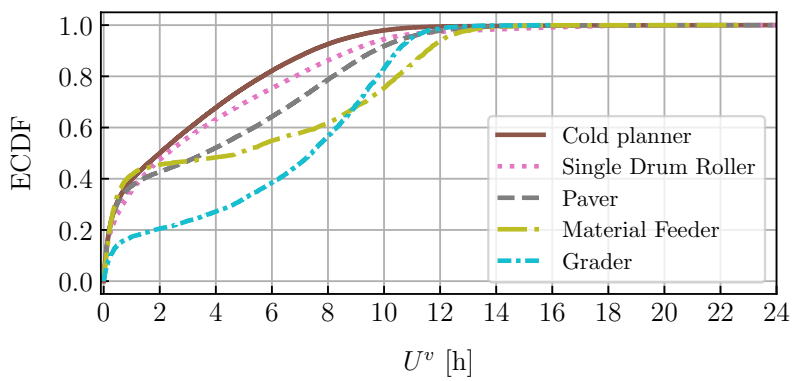

Figure 1: Empirical Cumulative Distribution Function of the number of daily utilisation hours per vehicles of different types (disregarding the inactive days).

The vehicle sensors and the machine control systems generate messages that are exchanged on the CAN bus at a high frequency (up to 100 Hz). Several on-board controllers acquire, collect, and summarise the raw data, sending aggregated data reports to a centralised repository every 10 minutes, via mobile broadband connections. For each vehicle the reports contain a set of data features describing the engine and vehicle statuses, e.g., fuel level and distance covered. They also include information about the time spent in four different engine duties, namely long idle, idle, moving/working, and high workload.

4.2. Preliminary data exploration

Data characterisation is instrumental in discovering similarities and differences among vehicle usage patterns. Hereafter, we will focus on the utilisation hours per day. For each vehicle we consider the whole 4-year dataset. Figure 1 shows the Empirical Cumulative Distribution Function (ECDF) of the daily utilisation hours separately for five different types. In an ECDF an arbitrary curve value $F(x)$ indicates the fraction of days in which the number of daily utilisation hours is less than or equal to x. The plot highlights the heterogeneity of the vehicle usage distributions across different types. For instance, graders are used more than 6 hours per day most of the times, whereas cold planners show opposite usage patterns, with a median usage of about 2 hours. Some vehicle types expose a long tail in the ECDF, meaning that they sometimes work up to 24 hours per day.

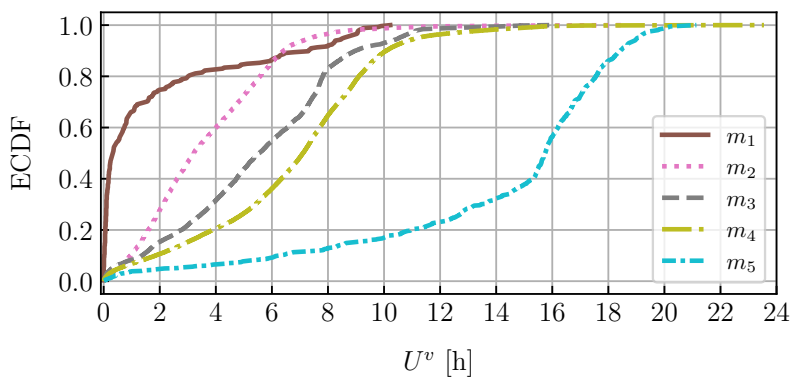

Figure 2: Empirical Cumulative Distribution Function of the number of daily utilisation hours per models of refuse compactors.

Within a specific vehicle type (i.e., refuse compactor), Figure 2 plots the ECDF of the vehicle utilisation levels separately for a subset of models, i.e., each curve aggregates the utilisation levels of all the refuse compactors belonging to a specific model. Also here, large differences are visible.

Figure 3 shows the series of daily utilisation levels (i.e., $U^v(t)$) for two representative vehicles. Vehicle v_1 has a daily utilisation level of about 10 hours, with a few inactive days every 10-15 working days. Conversely, vehicle v_2 is almost unused for several weeks (from t=0 to t=40 days) and then suddenly changes its usage pattern. This confirms the heterogeneity of the analysed data also under the perspective of the time left to maintenance. In a nutshell, vehicle usage patterns show different, non-stationary, and uncorrelated trends across models and types, suggesting to build separate regression models for each vehicle.

Moving to the time to next maintenance T^v, in Figure 4 we show the temporal evolution of the remaining time to the next maintenance for the two example vehicles. It exemplifies the maintenance cycles' turnover for different vehicles. When the value of $D^v(t)$ reduces to zero, it means that the vehicle needs to go to maintenance. Then a new maintenance cycle starts, the number of days left to maintenance is maximal and it monotonically decreases (one day for each day passed) until the next maintenance operation is carried out. Notice how v_1 has a first long cycle (221 days), while the others are more constant, with length between 65 and 105 days. It means that v_1 was underutilised at the beginning, so it took 221 for it to reach its utilization

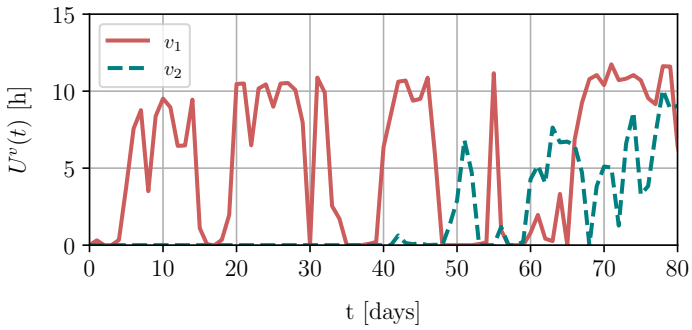

Figure 3: Example of time series of daily utilisation hours of two vehicles.

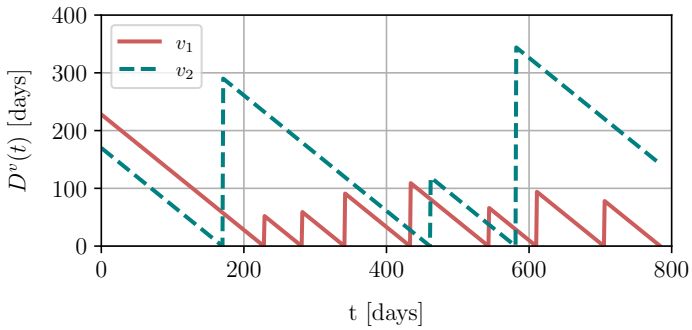

Figure 4: Maintenance cycle turnover: number of remaining days until the next maintenance (D^v) for two representative vehicles.

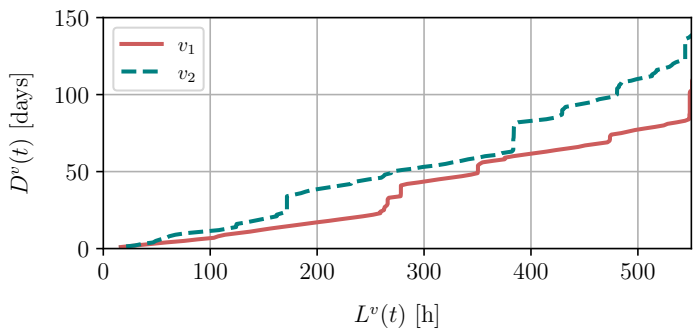

Figure 5: Utilisation time left to maintenance (L^v) vs. the number of days to maintenance (D^v, in days) for a single cycle of two representative vehicles.

maximum, before it had to go to maintenance, while afterwards it was used much more.

In Figure 5 we show the variation of the time left to maintenance ($D^v(t)$, expressed in days) with the utilisation time left for the next maintenance ($L^v(t)$, expressed in hours). As expected, a direct relationship holds. The utilization time left to maintenance is less than the actual time left to maintenance unless the vehicle works 24 hours per day until the day of the maintenance. Most of the time the utilisation rate is relatively constant and above zero. However, there are some vertical steps, corresponding to consecutive days on which the utilisation was null. This highlights the presence of low- or zero-utilisation periods and has a relevant impact on the target variable. Thus, predicting the correct target value could be challenging. Hopefully, it is unlikely to see long periods of zero-utilisation in the days approaching the deadline. This reinforces the motivations behind using $E^v_{MRE}(\widetilde{D})$ as error metric, which focuses on the values relatively close to the maintenance (see Section 3).

4.3. Data preparation and enrichment

We prepare the data for the machine learning process by applying standard preprocessing steps such as *data cleaning*, to handle missing values or minor inconsistencies in data (note that this step is performed upstream by the company during the data collection process), *normalization*, to make series values comparable with each other, and *aggregation*, to sample values

on a daily basis, and *enrichment*, to extend the raw data with contextual knowledge and additional related features.

Among the steps enumerated above, data enrichment deserves further details. Contextual information provides deeper insights into the actual ways in which the vehicles are used. For example, distinguishing between working days and holidays may be useful to determine the extent to which vehicles are likely to be used. Since all the vehicles are geo-referenced, holidays are computed according to the country where the vehicle is located.

In detail, in this work we considered the following features:

1. $U^v(t)$ [hours]: daily utilisation series of vehicle v.
2. $L^v(t)$ [hours]: series of the utilisation times left to the next maintenance operation for vehicle v.
3. Day type (working day or holiday).
4. Daily distance covered.
5. Daily fuel consumed.
6. Daily time spent in each of the engine duties.

Notice that the day type is known in advance for the target date as well. Such information is potentially relevant because it allows us to correlate future and past vehicle usage within specific day types (e.g., predict the utilisation hours of a vehicle on Sunday given that in the last 3 Sundays the past utilisation hours were zero).

5. Methodology

This section describes the methodologies used to address task A and task B, diversified by vehicle category (i.e., *new*, *semi-new*, and *old*) and by input data.

5.1. Task A: next-day utilisation level prediction

Univariate models. Task A entails predicting the utilisation level on the next day $U^v(t+1)$ based on the series of historical values $U^v(t)$, $U^v(t-1)$,..., $U^v(t-w+1)$ within a predefined time window of size w. We model the relation between the next-day utilisation level and the most recent utilisation levels observed within a period of duration w as an arbitrary regression function f_u:

$$U^v(t+1) = f_u(U^v(t), U^v(t-1), \ldots, U^v(t-w+1))$$

where $U^v(t+1)$ is the value of the target variable and $f(\cdot)$ is the prediction function we want to find.

Multivariate model. The prediction model is enriched, considering additional contextual features F_x^v describing different usage patterns associated with the vehicle v under analysis (see Section 4). Each feature F_x^v is described by the series of the historical values in the time window $F_x^v(t-w+1), F_x^v(t-w), \ldots, F_x^v(t-1), F_x^v(t)$. All feature values are assumed to be known within the considered time period $[t-w+1,t]$.[3] Section 4 details the list of contextual features we selected and included in the regression models. Formally speaking, the multivariate prediction model with contextualised information is the function f_m:

$$U^v(t+1) = f_m(U^v(t), \ldots, U^v(t-w+1), \ldots, F_x^v(t), \ldots, F_x^v(t-w+1), \ldots)$$

5.2. Task B: remaining time to next maintenance

To accomplish task B we learn a univariate regression function g_u to predict the number of days left to maintenance $D^v(t)$ for a given vehicle v based on the latest value of the daily utilisation time series L^v:

$$D^v(t) = g_u(L^v(t)) \tag{5}$$

Note that the correlation with the latest daily utilisation time left to the next maintenance is assumed to incorporate all the necessary information about the past vehicle usage patterns.

We try also to enrich the univariate regression model by considering the series of the historical daily utilisation levels $U^v(t-w+1), U^v(t-w), \ldots, U_x^v(t)$ within a size-w window time interval $[t-w+1,\, t-1]$. The goal is to estimate the following function g_m:

$$D^v(t) = g_m(L^v(t), U^v(t-1), \ldots, U^v(t-w)) \tag{6}$$

Note that in the latter case we do not explicitly include contextual features such as CAN bus-related, temporal, and spatial features since likely they just influence the daily utilisation series values.

[3]Note that the value of the temporal features (e.g., day of the week) are known even at time $t+1$.

5.3. Training strategy

For each vehicle v in the fleet we collect the historical usage data in a separate relational dataset. Each record stores the specific usage levels and contextual information acquired on a given day. Each record is also labelled a target variable according to the prediction task under analysis (next-day utilisation for task A, or remaining time to the next maintenance for task B).

To train the regression models we apply the following strategies. We first train both univariate and multivariate regression models on a portion $[t - TW + 1, t]$ of the historical data before day t, where TW is denoted as *training window*, and then test the models on the subsequent days (e.g., the next day $t + 1$ for task A). When the training phase requires a relatively long training period (e.g., for task B) we keep the training window fixed (70% of the data) and test the unique model on each of the subsequent days.[4] To do cross-validation (e.g., for task A), we vary the duration of the training window TW, using two alternative approaches: a sliding window and an expanding window strategy (Ratanamahatana et al., 2010). More specifically, by adopting a sliding window approach, we consider a fixed size TW for every training and test step. Hence, to learn the predictors we trust only the most recent vehicle usage data. Conversely, by using an expanding window approach the regression algorithm relies on the entire set of data available up to that point in time. Therefore, at an arbitrary time point t, $TW \in [t_0, t]$, where t_0 is the first data point (the first day for which we have data). The strategies are exemplified in Figure 6.

We performed the explained cross-validation procedure to tune parameters like the model window size w. Moreover, for task A we used many additional features (Section 4.3). To overcome the well-known curse of dimensionality problem (Zaki & Meira, 2020) we focus the training phase of the regression functions on the most discriminating features. Separately for each vehicle, we select the top-k most relevant features. To this aim, we first apply the statistical F-test (James et al., 2014), which estimates the degree of linear dependency between the target class and each of the candidate features. Then, we select the top-k most correlated features by using the implementation available in the Scikit-Learn library (Pedregosa et al., 2011).

[4]For semi-new vehicles we include only usage data related to the first cycle in the training set.

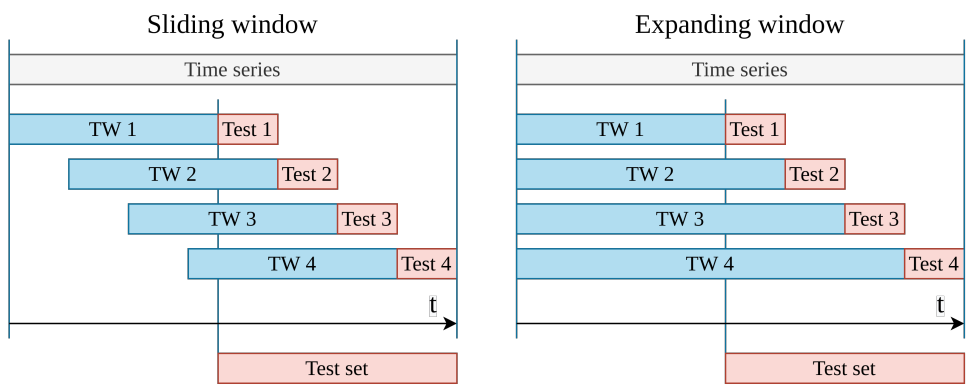

Figure 6: Sliding vs. expanding window strategies.

5.4. Analysis of old vehicles

By construction, old vehicles have a large amount of historical data to learn vehicle-specific regression models (see Section 3). Hence, we can train ad-hoc regression models to predict the value of the target class.

As shown in Section 4, the characteristics of the vehicles in the fleet are rather heterogeneous. To explore the added value provided by ML algorithms in accomplishing tasks A and B, we compare the performance of regression algorithms (both univariate and multivariate) with the ones of the following baseline methods:

Baseline method for task A. It assumes that the utilisation level for a vehicle in the next day will be the same as those observed in the current day.

$$U^v_{BL}(t) = U^v(t-1) \tag{7}$$

Baseline method for task B. It assumes that the utilisation level in the future will follow the same trend as observed in the past. Firstly, it estimates the average utilisation level of vehicle v on the training set with size TW.

$$AVG^v = \frac{\sum_{i=1}^{TW} U^v(t)}{TW} \tag{8}$$

Next, assuming a stationary vehicle usage, it predicts the number of remaining days until the next maintenance for v as follows.

$$D^v_{BL}(t) = \frac{L^v(t)}{AVG^v} \tag{9}$$

5.5. Analysis of new and semi-new vehicles

To analyse semi-new and new vehicles, we have to address the lack of historical usage data, which hinders the training of per-vehicle regression models. The key idea is to reuse the historical data available for similar vehicles (whenever feasible).

We present three different strategies. For each strategy we differentiate the approach used for *semi-new* vehicles, which have completed at least half of the first maintenance cycle (but a full cycle has not terminated yet), from those applied to *new* vehicles, for which very few samples are available (the minimum depends on the used model).

Baseline strategy. As in Section 5.4, to accomplish task A, it uses the last utilisation level as in equation (7), while for task B it computes the average utilisation level and predict the next maintenance as in equation (9).

Motif-driven ML strategy. This strategy focuses on first identifying the vehicles that show the most similar usage trends, and then training the prediction models only on usage data acquired from those vehicles.

The problem of finding pairs of vehicles showing highly similar usage patterns can be reformulated as the motif discovery problem from time series data (Mueen et al., 2009). Let $V=\{v_1, v_2, \ldots, v_n\}$ be a set of candidate old vehicles. We aim at identifying the vehicles in V exhibiting a usage level series, that is similar to those of the target vehicle v. For the target vehicle v, let U^v be the series of available utilisation levels of v. The motif that best approximates U^v is the sub-series $U^{v_B}_{motif}$ with $v_B \in V$ such that:

$$U^{v^B}_{motif} = \arg\min_{U^{v^*}_{sub}} \mathrm{dist}(U^v, U^{v^*}_{sub})$$

where $U^{v^*}_{sub}$ is an arbitrary sub-series of U^{v^*} belonging to the first cycle and whose length is equal to those of U^v and $dist(\cdot)$ is a distance measure.

The motif-driven strategy trains the regression model on the training dataset including only the best approximated motif of U^v_{hc} within the vehicles of the same type. To limit the complexity of the motif discovery process, we greedily explore the space of candidate utilisation level sub-series $U^{v^*}_{sub}$ by considering three uniformly distributed samples per vehicle $v^* \in V$. The pairwise time series distance $dist(\cdot)$ is computed as the point-wise average distance AVG^v. Note that the proposed strategy can be straightforwardly extended by considering alternative similarity measures, e.g., (Neamtu et al.,

2018), as well as more efficient, variable-length motif discovery algorithms, e.g., (Linardi et al., 2018).

The motif-driven strategy for Task B is applicable to only semi-new vehicles since it requires the availability of some historical usage data related to (at least) the first half of the maintenance cycle.

Type-specific ML model. Here we combine into a unique training dataset the usage data acquired in the first maintenance cycle by *all* the vehicles of the same type. The goal is to train a unified regression model tailored to each specific vehicle type. The model can be applied to both new and semi-new vehicles and trains on historical data acquired from old vehicles. We performed this approach for task B only.

5.6. Regression algorithms

We train and test both linear and non-linear regression models. Non-linear models are potentially able to capture more complex and possibly non-stationary usage trends. As a drawback, the complexity of non-linear models is typically higher than that of linear ones.

To solve the prediction tasks we employ the following regression algorithms: (i) Linear Regression (LR), which fits a linear function minimising the residual sum of squares between the true target value and the predicted one. (ii) Support Vector Regressor (SVR), which separates points in multi-dimensional space with a hyper-plane. (iii) Random Forest Regressor (RF), which is an ensemble method that combines the predictions of multiple decision trees. (iv) Gradient Boosting (GB), which combines decision trees by applying a boosting strategy. A more detailed description of the aforesaid algorithms is given in Zaki & Meira (2020). Due to the limited amount of training data, in the experimental analyses we neglect the categories of algorithms that are most sensitive to data overfitting (e.g., Neural Networks).

We evaluate algorithm performance in terms of both accuracy of the predictions and efficiency. The latter is expressed by considering the training time, which is among the most relevant factors influencing the usability of ML-based decision support systems. For each algorithm we explore hyper-parameter optimisation via grid search to find the configuration settings that best fits the input data distribution. For each algorithm we also identify a recommended configuration setting, which allows us to achieve fairly high performance for all the tested cases.

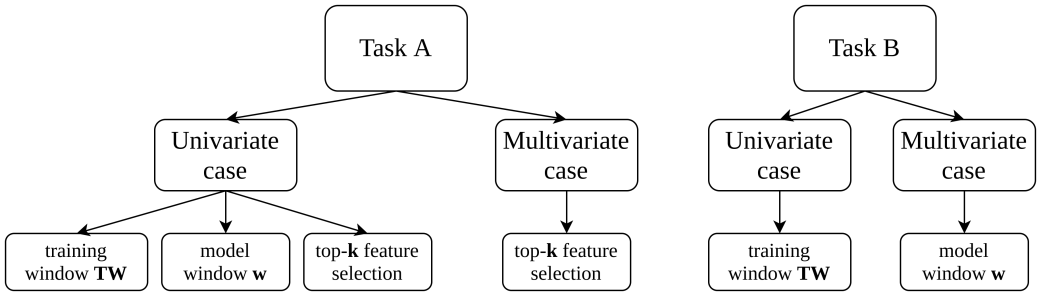

Figure 7: A schema of all the parameters we tune in the experiments for the two tasks.

6. Experimental results

We conduct an extensive experimental campaign on real vehicle usage data acquired by the telematics service provider, analysing both tasks A (Section 6.1) and B (Section 6.2), their performance for the different types of vehicles (old, new and semi-new) and the respective computational complexity (Section 6.3). To find the best-performing parameter configuration, we tune a few parameters for both tasks. A schema of them is depicted on Figure 7. Note that the tuning is done for old vehicles, so that the best-performing models can be used for semi-new and new vehicles.

6.1. Performance analysis of next-day utilization level predictors (task A)

In this section we analyse the performance of task A for both old and new vehicles. We test different configuration parameters in terms of kinds of features (univariate and multivariate case), possible feature selection, and sizes of the training window TW and model window w and tune the hyper-parameters of the algorithms. Hereafter, we will focus on the data-richest vehicle type, i.e., refuse compactor.

First we show the best obtained results for each algorithm. For this test we employ the configuration setting recommended later on (see Sections 6.1.1-6.1.3), i.e., multivariate scenario with top-12 feature selection with $w = 30$ and $TW = 140$. The box-plot depicted in Figure 8 shows the distribution of the percentage errors $E^v_{\%}$ achieved by the selected algorithms, over the considered (60) old vehicles. For each algorithm the box summarises the percentage error distribution by representing the interquartile range IQR (Q3-Q1), the whiskers span (1.5·IQR), the median value (the orange line), and the mean (the triangle).

All the approaches outperformed the baseline approach. They allowed us to roughly halve the average percentage error for the old vehicles. All the ML-based strategies achieved similar performance, except for LR whose results show a significantly higher error due to the presence of non-linear trends in the data.

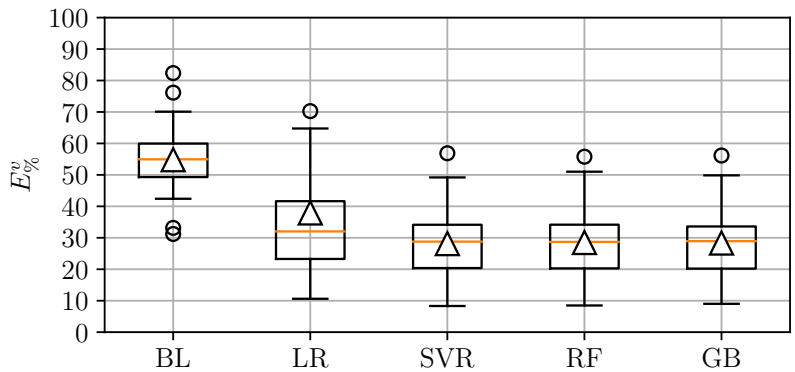

Figure 8: Algorithm comparison for task A in terms of percentage error over old vehicles (multivariate scenario with top-12 feature selection with $w = 30$ and $TW = 140$).

Focusing on new and semi-new vehicles, the box-plot in Figure 9 shows the corresponding percentage errors, using the motif-driven strategy. Again, the model used exploits a multivariate scenario with top-12 feature selection with $w = 30$ and $TW = 140$. Compared to the old vehicle scenario, in the new/semi-new context the error shows a significantly higher variance with a 15% increase in median value. This means that training on other vehicles will not perform as good as training on the vehicle itself, but this is the best thing we can do without historical data. Notice also how the lack of data impacts the accuracy of predictions, which are now much more variable than for old vehicles. Still, ML-based approaches outperform the baseline strategy (BL) by around 30%.

6.1.1. Effect of the training window size

For old vehicles, we observe the impact of the size of the training sliding window TW from 60 days to all the available past data (the latter case corresponds to the expanding window strategy). We start here with the simple univariate scenario without feature selection and $w = 40$. Figure 10 shows the variation of the median percentage error for all the tested algorithms,

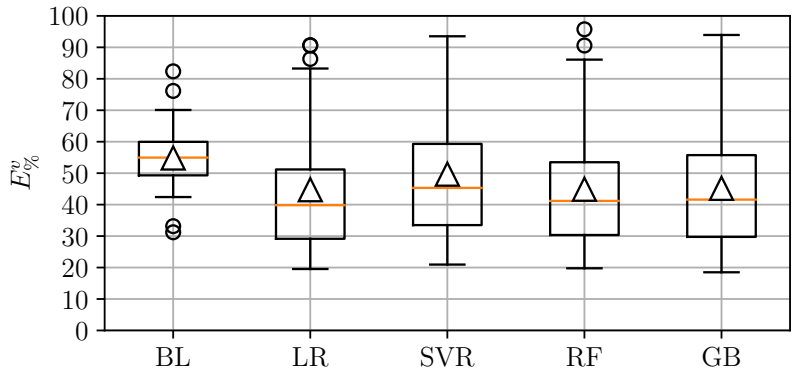

Figure 9: Algorithm comparison for task A in terms of percentage error over the new and semi-new vehicles (motif-driven strategy in a multivariate scenario with top-12 feature selection with $w = 30$ and $TW = 140$).

over the old vehicles. The performance of the algorithms improves by increasing the training window size, with the baseline approach (BL) that is obviously not dependant by the window size. An increase of the window size results in a sizeable and non-linear complexity increase of the training phase, we decide to set the window size to 140 to achieve a satisfactory trade-off between effectiveness and efficiency of the proposed solution.

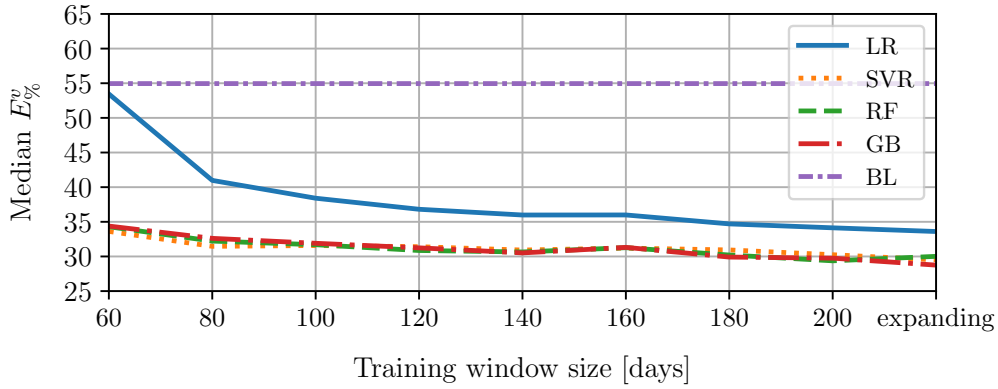

Figure 10: Task A: effect of the training window size on the median percentage error (old vehicles, univariate scenario without feature selection and with $w = 30$).

6.1.2. Effect of the feature selection step

We separately analyse two complementary effects related to the selection of the most appropriate training data features: (i) the inclusion in the training set of historical series values acquired in several past days (model window w) and (ii) the impact of the correlation-based feature selection strategy described in Section 5.3 (top-k feature selection). Here, for simplicity, we still consider the univariate case, with $TW = 140$.

Regarding point (i), Figure 11 shows the variation of the median percentage error while including in the model window size w a number of past days ranging from 1 (only the last day) to 51 (more than 7 weeks before). The results show that by using less than the values in the past 9 days the predictors appeared to degrade in performance, independently of the tested algorithm, whereas by including more than 21 past days in the model window size w, the models produced either marginal improvements (e.g., for RF and GB) or even worse percentage errors (for LR and SVR), due to the weak correlation with very old data and the higher complexity in fitting the model. In summary, taking into account the influence of short-term trends observed in the last three weeks shows to be sufficient to achieve relatively high prediction performance.

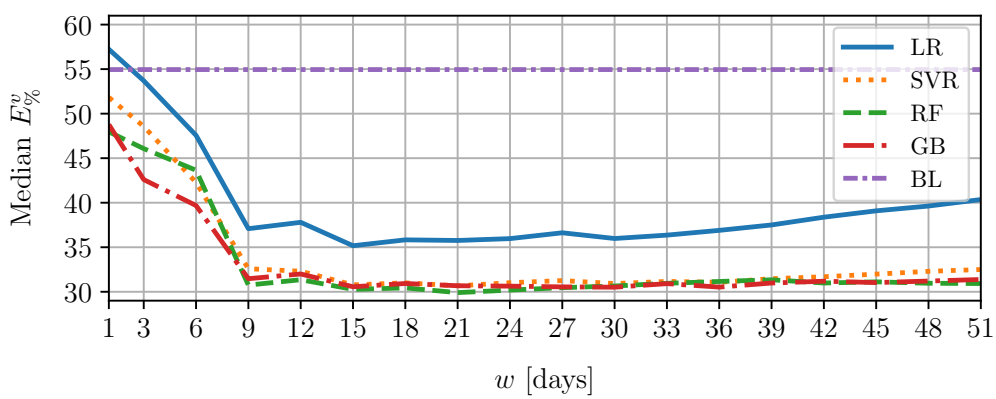

Figure 11: Task A: Effect of the number of past days considered in the predictive model, i.e., the model window size w (univariate scenario without feature selection with $TW = 140$).

Regarding point (ii), Figure 12 shows the trend of the median percentage

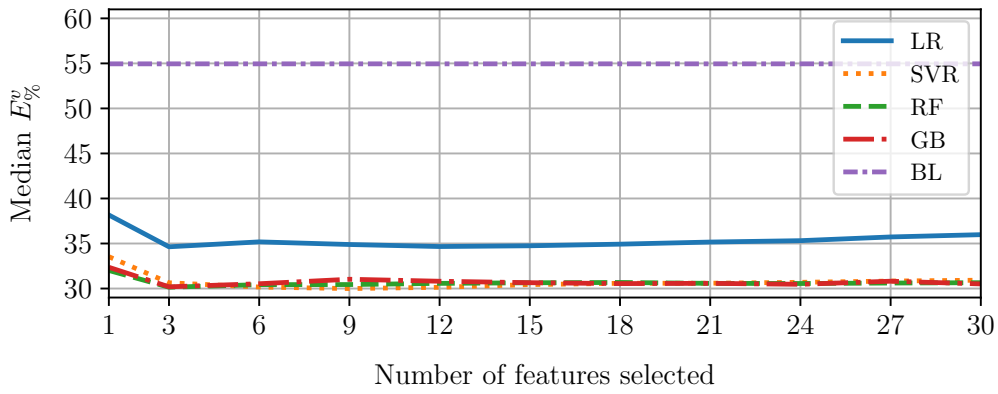

Figure 12: Task A: Effect of top-k feature selection on the percentage error (univariate case, with $w = 30$ and $TW = 140$).

error by varying the number of features selected according to their importance as given by the F-test statistics (from 1 feature to 30). The best results are achieved by selecting from 3 to 12 features. Interestingly, in the majority of the cases the selected features refer to the utilisation levels of the previous three days or the same weekday in the previous few weeks. This confirms the primary importance of detecting short-term trends in utilisation series, and suggests a weekly seasonality.

6.1.3. Multivariate scenario

We now test the regression algorithm in the multivariate scenario. We consider several combinations of input features, such as, the distance covered by the vehicle, the fuel consumed, the day type (working day or holiday), and the time spent in various engine duty states. Here we fix $w = 30$ and $TW = 140$.

The results show that:

- The multivariate models trained using the past usage level and the day type is, on average, the best in terms of average percentage error.
- The feature selection step was effective even in the multivariate case, with the optimal number of selected features here being 12 (instead of 3 in the univariate case), since there are more features to begin with.

In order to evaluate the significance of the performance improvements achieved by integrating specific contextual features in the multivariate model, we apply the statistical Z test on the observed prediction errors. To this end, we first split the runs not including a target feature (e.g., day type) from the corresponding ones including it (more than 6000 runs per split to deal with a sufficiently large population of samples). Next, we carry out the Z test on the corresponding percentage error distributions to verify the hypothesis that the two population means are different assuming that the variances are known and the distributions are normal. Notice that the latter assumption trivially holds for the average percentage errors from the central limit theorem. According to the test outcomes, the day type feature improves the algorithm performance at 95% significance (Z statistic 2.37, p-value 0.018). This confirms the importance of exploring the multivariate scenario.

6.2. Performance of the time to next maintenance predictors (task B)

In this section we analyse the performance of Task B for old, new and semi-new vehicles. Here we test different configuration settings, with particular attention to the window size w which, according to the previous experiments, has shown to be particularly relevant. Following the domain experts' recommendations, in our the experiments we set the maintenance cycle duration to 555 hours (23 days), which is deemed as a value suitable for Refuse compactors. Thus, if a vehicle has worked for 555 hours, during any number of calendar days, it then has to go to maintenance. Note that this is an input parameter that can be approximated based on domain-specific knowledge and easily adapted to the current vehicle and usage context under analysis.

Table 1: Task B: Mean residual error $E_{MRE}(\{1,\ldots,29\})$ computed over the last 29 days before the maintenance using different training window settings (univariate case).

Algorithm	Trained on all data $E_{MRE}(\{1,\ldots,29\})$	Trained on $D=\{1,\ldots,29\}$ $E_{MRE}(\{1,\ldots,29\})$
BL	20.2	20.2
LR	26.1	10.8
SVR	13.3	6.1
RF	6.9	2.4
XGB	10.9	5.6

We start from the univariate case. Table 1 reports the mean residual error computed over the last 29 days before the maintenance $E_{MRE}(\{1,\ldots,29\})$ by

training each algorithm using two different training window settings. Specifically, the left hand-side column reports the error when using all the available training data, whereas the right hand-side one reports the outcomes produced by using only the last 29 days. The latter training strategy achieves significantly better performance: the error is 59% lower than in the former case using LR, 54% lower using SVR, 65% lower using RF, and 48% lower using XGB. Here RF achieves the best results, with an average relative error of only 2.4 days, when trained over the last 29 days.

6.2.1. Effect of the feature selection step

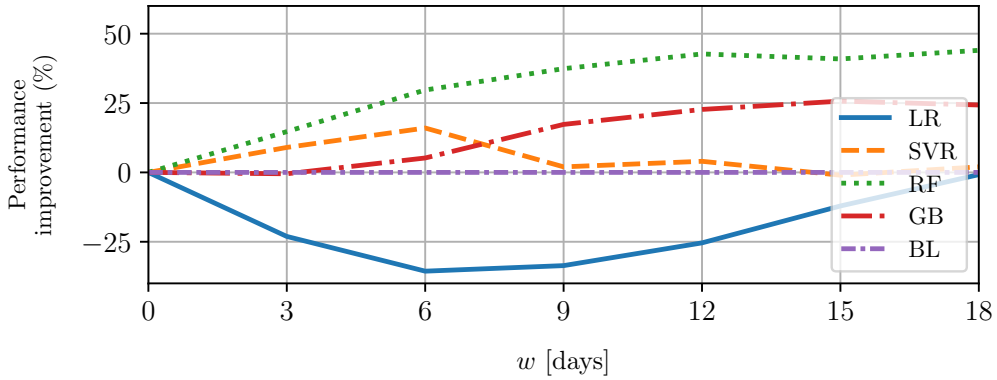

Figure 13: Task B: Effect of the number of past days w considered in the predictive model (% performance improvement is shown for each algorithm).

Table 2: Task B: Optimal window size setting and corresponding mean residual error for each algorithm (multivariate case).

Algorithm	Best window w	$E_{MRE}(\{1,\ldots,29\})$
BL	0	20.2
LR	0	10.8
SVR	6	5.2
RF	18	1.3
XGB	12	4.2

Figure 13 plots the performance variation, in percent, when increasing the window size, for each algorithm. Positive/negative variations indicate a

decrease/increase of the prediction error shown in Table 1 with respect to the baseline (w=0). Limiting to the results achieved by the best performing algorithms (i.e., RF and XGB), including additional past observations in the training data improves the prediction. They both reach a steady state when more than 15 past days are considered. The performance improvements are substantial: 44% and 25%, respectively. Further extending the training window size does nor produce any relevant improvements, probably due to the presence of noise or out-of-date trends. Table 2 summarises the best obtained $E_{MRE}(\{1, \ldots, 29\})$ for each algorithm. Ensemble methods (i.e., XGB and RF) are overall the best ones, consistently with the results achieved for task A.

Finally, Figure 14 plots the variation of $E_{MRE}(\widetilde{D})$, separately for each algorithm, in each of the last 29 days before maintenance. We recall that $\widetilde{D}$ indicates the number of remaining days until the next maintenance *independently of the current maintenance cycle* (i.e., each error is averaged over all cycles). Obviously, the closer the next maintenance, the lower $\widetilde{D}$, and the lower the error. All the ML-based approaches perform significantly better than the baseline. RF achieves very promising errors (average error around 2.4 days) even 29 days before the deadline, making the prediction system effective to support domain experts in scheduling maintenance actions in advance.

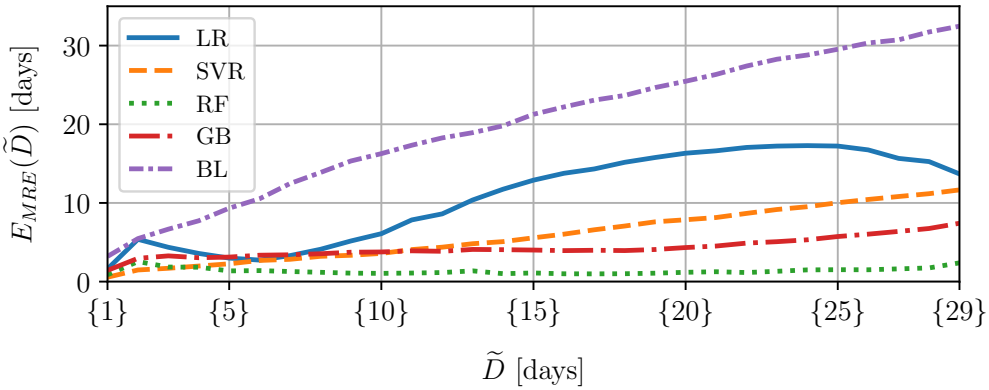

Figure 14: Task B: Mean residual error computed for different $\widetilde{D}$ ranging from 1 to 29 days.

Table 3: Results on semi-new and new vehicles for task B (motif-driven and type specific).

Algorithm	Semi-new vehicles $E_{MRE}(\{1,\ldots,29\})$	New vehicles E_{Global}
$BaseLine$	34.9	-
$LR_{Motif-driven}$	4.9	-
$SVR_{Motif-driven}$	6.2	-
$RF_{Motif-driven}$	2.9	-
$XGB_{Motif-driven}$	5.3	-
$LR_{Type-spec}$	5.1	27.2
$SVR_{Type-spec}$	8.8	27.8
$RF_{Type-spec}$	3.2	30.1
$XGB_{Type-spec}$	4.2	17.9

6.2.2. Results for new and semi-new vehicles

Analogously to what was previously done for task A, we deeply analyse the performance of the proposed approach for semi-new and new vehicles. The results are summarised in Table 3). On semi-new vehicles the baseline strategy performs relatively poorly, with a mean relative error of 34.9 days over the last 29 days. Conversely, all the ML-based solutions (both type-based and motif-driven) improve the baseline performance at least 5 times (average error ranging between 2.9 and 8.8 days). Thus, exploiting regression algorithms provides clear advantages. As in the old vehicle scenario, Random Forest turns out to be the best performing predictor. Interestingly, motif-driven approaches perform on average 10% better than the corresponding type-specific ones (e.g., $E_{MRE}(\{1,\ldots,29\})$ $RF_{Motif-driven}$ 2.9 vs. $RF_{Type-spec}$ 3.2). Hence, exploring similar vehicle usage time series data helps to forecast the upcoming usage trend.

On new vehicles, neither baseline nor motif-driven strategies are applicable. Hence, we explore the use of type-specific models. $XGB_{Type-spec}$ performs best with an average error equal to 17.9 days. The error is quite large three weeks before the maintenance then it becomes similar to those achieved by the baseline method (LV) on old vehicles (see Table 2).

6.3. Model computation time

We analyse the complexity of the proposed approaches in terms of execution time taken by the overall analytical process. The experiments were performed on an Intel(R) Core(TM) i7-8550U CPU with 16 GB of RAM running Ubuntu 18.04 server. The programming language used was Python.

The machine learning pipeline consists of three main phases: (i) Data preparation, (ii) Model learning, and (iii) Model application. Phase (ii) turns out to be the most time-consuming one for all the algorithms. Specifically, for task A, using the best parameters and feature selection, the training time of one model varies between less than a second for simpler models (0.38 s and 0.49 s on average for LR and SVR, respectively) and a few seconds seconds for the more complex RF (3.4 s). Surprisingly, training GB takes only 0.64 s on average. Note that these numbers correspond to training one model, while in our methodology we train as many models as the amount of data we have for one vehicle, updating each day. Similarly, for task B, with best parameters and feature set, the average training time on a single vehicle is, on average, 30.4 s for XGB and 8.1 s for RF. LV, LR, and SVR respectively take 2.5 s, 3.8 s, and 2.8 s.

7. Conclusions and future works

The paper describes an industrial case study focused on the application of machine learning techniques to CAN bus data acquired from construction vehicles. In order to support fleet managers, we aimed at predicting the next-day utilisation level and the remaining days to the next preventive maintenance, for each vehicle in the fleet. We addressed two open research issues, i.e., the lack of historical data, which hinders the training of machine learning models, and the high heterogeneity of fleet vehicles, which limits the generality of the proposed solutions.

The experimental results, achieved on real vehicle usage data collected in various construction sites, show that:

- Linear regression models perform better than baseline methods, but still suffer from the high variability of the vehicle usage patterns (see, for example, Figure 8).
- Ensemble methods like Random Forest and Gradient Boosting are effective in performing per-vehicle predictions, provided that a sufficient amount of training data are available (see, for example, Figure 11).
- Contextual features describing temporal vehicle usage recurrences are helpful to improve univariate model performance (see Section 6.1.3).
- When there is a lack of historical data it is worth leveraging the similarity between the vehicle limited usage time series and those of similar

old vehicles. Tailoring the learning process to vehicles showing approximated motifs in the first maintenance cycle shows 10% average error reduction compared to including all vehicles of the same type.

- The company involved in this industrial research project has put the application under deployment. On the one hand, this demonstrates the usability of the proposed solution. On the other hand, it leaves room for further improvements and extensions (see, for example, Figure 9 and Table 3).

For future research work we are integrating multi-scale and multi-dimensional vehicle usage data, transferring the models learned from on-road vehicles in this new, more challenging scenarios, and specialising the current knowledge on vehicle workload states with both domain-specific information and data-driven, descriptive models (e.g., (Buccafusco et al., 2021; Fugiglando et al., 2017)).

References

Almer, H. (2015). *Machine learning and statistical analysis in fuel consumption prediction for heavy vehicles*. Master's thesis KTH, School of Computer Science and Communication (CSC).

Alonso de Armiño, C., Manzanedo, M. Á., & Herrero, Á. (2019). Studying road transportation demand in the spanish industrial sector through k-means clustering. In M. Graña, J. M. López-Guede, O. Etxaniz, Á. Herrero, J. A. Sáez, H. Quintián, & E. Corchado (Eds.), *International Joint Conference SOCO'18-CISIS'18-ICEUTE'18* (pp. 387–396). Cham: Springer International Publishing.

Barreto, L., Amaral, A., & Pereira, T. (2017). Industry 4.0 implications in logistics: an overview. *Procedia Manufacturing*, *13*, 1245–1252.

Buccafusco, S., Megaro, A., Cagliero, L., Vaccarino, F., Salvatori, L., & Loti, R. (2021). Profiling industrial vehicle duties using can bus signal segmentation and clustering. volume 2841. URL: `https://www.scopus.com/inward/record.uri?eid=2-s2.0-85103491379&partnerID=40&md5=d5ff8b28a34784c40a9975167f2a75ee`.

Caapraz, A. G., Ozel, P., Azevkli, M., & Beyca, A. F. (2016). Fuel consumption models applied to automobiles using real-time data: A comparison of statistical models. *Procedia Computer Science*, *83*, 774 – 781. URL: http://www.sciencedirect.com/science/article/pii/S1877050916301995. doi:https://doi.org/10.1016/j.procs.2016.04.166. The 7th International Conference on Ambient Systems, Networks and Technologies (ANT 2016) / The 6th International Conference on Sustainable Energy Information Technology (SEIT-2016) / Affiliated Workshops.

Crainic, T., & Laporte, G. (1998). *Fleet Management and Logistics*. Centre for Research on Transportation. Springer US.

Dalzochio, J., Kunst, R., Pignaton, E., Binotto, A., Sanyal, S., Favilla, J., & Barbosa, J. (2020). Machine learning and reasoning for predictive maintenance in industry 4.0: Current status and challenges. *Computers in Industry*, *123*, 103298. URL: http://www.sciencedirect.com/science/article/pii/S0166361520305327. doi:https://doi.org/10.1016/j.compind.2020.103298.

Delgado, O. F., Clark, N. N., & Thompson, G. J. (2012). Heavy duty truck fuel consumption prediction based on driving cycle properties. *International Journal of Sustainable Transportation*, *6*, 338–361. doi:10.1080/15568318.2011.613978.

Fugiglando, U., Santi, P., Milardo, S., Abida, K., & Ratti, C. (2017). Characterizing the "driver dna" through can bus data analysis. (pp. 37–41). doi:10.1145/3131944.3133939.

Halim, Z., Kalsoom, R., & Baig, A. R. (2016). Profiling drivers based on driver dependent vehicle driving features. *Applied Intelligence*, *44*, 645–664. URL: https://doi.org/10.1007/s10489-015-0722-6. doi:10.1007/s10489-015-0722-6.

Hellstrom, E., Ivarsson, M., Aslund, J., & Nielsen, L. (2009). Look-ahead control for heavy trucks to minimize trip time and fuel consumption. *Control Engineering Practice*, *17*, 245 – 254. URL: http://www.sciencedirect.com/science/article/pii/S0967066108001251. doi:https://doi.org/10.1016/j.conengprac.2008.07.005.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2014). *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated.

Johansson, K. H., Törngren, M., & Nielsen, L. (2005). Vehicle applications of controller area network. In D. Hristu-Varsakelis, & W. S. Levine (Eds.), *Handbook of Networked and Embedded Control Systems* (pp. 741–765). Boston, MA: Birkhäuser Boston. URL: https://doi.org/10.1007/0-8176-4404-0_32. doi:10.1007/0-8176-4404-0_32.

Li, Q., Meng, S., Wang, S., Zhang, J., & Hou, J. (2019). CAD: command-level anomaly detection for vehicle-road collaborative charging network. *IEEE Access*, *7*, 34910–34924. URL: https://doi.org/10.1109/ACCESS.2019.2904047. doi:10.1109/ACCESS.2019.2904047.

Linardi, M., Zhu, Y., Palpanas, T., & Keogh, E. (2018). Matrix profile x: Valmod - scalable discovery of variable-length motifs in data series. In *Proceedings of the 2018 International Conference on Management of Data* SIGMOD '18 (p. 10531066). New York, NY, USA: Association for Computing Machinery. URL: https://doi.org/10.1145/3183713.3183744. doi:10.1145/3183713.3183744.

Markudova, D., Baralis, E., Cagliero, L., Mellia, M., Vassio, L., Amparore, E., Loti, R., & Salvatori, L. (2019). Heterogeneous industrial vehicle usage predictions: A real case. In P. Papotti (Ed.), *Proceedings of the Workshops of the EDBT/ICDT 2019 Joint Conference, EDBT/ICDT 2019, Lisbon, Portugal, March 26, 2019*. CEUR-WS.org volume 2322 of *CEUR Workshop Proceedings*. URL: http://ceur-ws.org/Vol-2322/DARLIAP_13.pdf.

Mishra, S., Vassio, L., Cagliero, L., Mellia, M., Baralis, E., Loti, R., & Salvatori, L. (2020). Machine learning supported next-maintenance prediction for industrial vehicles. In A. Poulovassilis, D. Auber, N. Bikakis, P. K. Chrysanthis, G. Papastefanatos, M. A. Sharaf, N. Pelekis, C. Renso, Y. Theodoridis, K. Zeitouni, T. Cerquitelli, S. Chiusano, G. Vargas-Solar, B. Omidvar-Tehrani, K. Morik, J. Renders, D. Firmani, L. Tanca, D. Mottin, M. Lissandrini, & Y. Velegrakis (Eds.), *Proceedings of the Workshops of the EDBT/ICDT 2020 Joint Conference, Copenhagen, Denmark, March 30, 2020*. CEUR-WS.org volume 2578 of *CEUR Workshop Proceedings*. URL: http://ceur-ws.org/Vol-2578/DARLIAP9.pdf.

Mohamed, E., Jafari, P., Siu, M.-F. F., & AbouRizk, S. (2017). Data-driven simulation-based model for planning roadway operation and maintenance projects. In *Proceedings of the 2017 Winter Simulation Conference* WSC 17. IEEE Press.

Morariu, C., Morariu, O., Rileanu, S., & Borangiu, T. (2020). Machine learning for predictive scheduling and resource allocation in large scale manufacturing systems. *Computers in Industry*, *120*, 103244. URL: http://www.sciencedirect.com/science/article/pii/S0166361519311595. doi:https://doi.org/10.1016/j.compind.2020.103244.

Mueen, A., Keogh, E. J., Zhu, Q., Cash, S., & Westover, M. B. (2009). Exact discovery of time series motifs. In *Proceedings of the SIAM International Conference on Data Mining, SDM 2009, April 30 - May 2, 2009, Sparks, Nevada, USA* (pp. 473–484). SIAM. URL: https://doi.org/10.1137/1.9781611972795.41. doi:10.1137/1.9781611972795.41.

Neamtu, R., Ahsan, R., Rundensteiner, E. A., Sarkozy, G., Keogh, E., Dau, H. A., Nguyen, C., & Lovering, C. (2018). Generalized dynamic time warping: Unleashing the warping power hidden in point-wise distances. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)* (pp. 521–532). IEEE.

Nguyen, T. T., & Wilson, B. G. (2010). Fuel consumption estimation for kerbside municipal solid waste (msw) collection activities. *Waste Management & Research*, *28*, 289–297. doi:10.1177/0734242X09337656.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Perrotta, F., Parry, T., & Neves, L. C. (2017). Application of machine learning for fuel consumption modelling of trucks. In *2017 IEEE International Conference on Big Data, BigData 2017, Boston, MA, USA, December 11-14, 2017* (pp. 3810–3815). URL: https://doi.org/10.1109/BigData.2017.8258382. doi:10.1109/BigData.2017.8258382.

Rashidnejad, M., Ebrahimnejad, S., & Safari, J. (2018). A bi-objective model of preventive maintenance planning in distributed systems considering vehicle routing problem. *Computers & Industrial Engineering*, *120*, 360 – 381.

Ratanamahatana, C. A., Lin, J., Gunopulos, D., Keogh, E. J., Vlachos, M., & Das, G. (2010). Mining time series data. In *Data Mining and Knowledge Discovery Handbook, 2nd ed.* (pp. 1049–1077). Springer Publishing Company, Incorporated. URL: `https://doi.org/10.1007/978-0-387-09823-4_56`. doi:`10.1007/978-0-387-09823-4\_56`.

Robert, E., Berenguer, C., Bouvard, K., Tedie, H., & Lesobre, R. (2018). Joint dynamic scheduling of missions and maintenance for a commercial heavy vehicle: value of on-line information. *IFAC-PapersOnLine*, *51*, 837 – 842.

Wickramanayake, S., & Bandara, H. M. N. D. (2016). Fuel consumption prediction of fleet vehicles using machine learning: A comparative study. *2016 Moratuwa Engineering Research Conference (MERCon)*, (pp. 90–95).

Zaki, M. J., & Meira, W., Jr (2020). *Data Mining and Machine Learning: Fundamental Concepts and Algorithms*. (2nd ed.). Cambridge University Press. doi:`10.1017/9781108564175`.

Zeng, W., Miwa, T., Wakita, & Morikawa, T. (2015). Exploring trip fuel consumption by machine learning from gps and can bus data. *Journal of the Eastern Asia Society for Transportation Studies*, *11*, 906–921. doi:`10.11175/easts.11.906`.

Zhang, W., Yang, D., & Wang, H. (2019). Data-driven methods for predictive maintenance of industrial equipment: A survey. *IEEE Systems Journal*, *13*, 2213–2227.

Zhou, M., Jin, H., & Wang, W. (2016). A review of vehicle fuel consumption models to evaluate eco-driving and eco-routing. *Transportation Research Part D: Transport and Environment*, *49*, 203 – 218. URL: `http://www.sciencedirect.com/science/article/pii/S1361920916306009`. doi:`https://doi.org/10.1016/j.trd.2016.09.008`.